Curvature based updates for optimization problems on a hypersurface

Jochen Trumpf
Department of Information Engineering
The Australian National University
Canberra, ACT 0200, Australia
and National ICT Australia Ltd.*
Jochen.Trumpf@anu.edu.au

Jonathan Manton
Department of Information Engineering
The Australian National University
Canberra, ACT 0200, Australia
Jonathan.Manton@anu.edu.au

Robert Mahony
Department of Engineering
The Australian National University
Canberra, ACT 0200, Australia
Robert.Mahony@anu.edu.au

***Abstract*— We propose a new class of algorithms for the optimization of a smooth cost function over a smooth hypersurface. In each step of the algorithm the hypersurface is approximated by a quadric, the optimization problem is solved on the quadric and the result projected back onto the hypersurface. We illustrate with an example that this approach can be computationally feasible.**

***Keywords*— geometric optimization, hypersurface**

I. INTRODUCTION

Recently there has been significant interest in the development of efficient optimization algorithms for a class of constrained optimization problems where the constraint set is a smooth matrix manifold [5], [1]. The new algorithms are motivated by a range of linear algebra problems involving the factorisation of matrices and determination of invariant subspaces that can be reformulated as constrained optimization algorithms [4]. The matrix manifolds considered are often matrix groups and have a Lie-group structure, or are quotients of Lie-groups leading to homogeneous or even symmetric space structures. The main contribution of the last few years of work has been to apply the modern theory of differential geometry to these problems to provide a means to apply the tools and techniques of unconstrained optimization such as Newton methods and trust region methods to the solution of the optimization problem by exploiting the intrinsic geometry of the underlying constraints.

There are two fundamental assumptions driving efficacy of these algorithms:

1) The constraint set is highly symmetric.
2) The cost function is simple.

The symmetry of the constraint set is usually a natural consequence of the algebraic structure of the underlying problem. The cost functions considered are often simple quadratics such as the Rayleigh quotient $\phi(X) = \mathrm{tr}(X^T A X)$, or even a linear cost such as the moment map $\phi(X) = \mathrm{tr}(NX)$ considered in the seminal work on the Toda lattice [6], [9], [11] leading to work on the double bracket flow [7], [2], [3].

*National ICT Australia Limited is funded by the Australian Government's Department of Communications, Information Technology and the Arts and the Australian Research Council through Backing Australia's Ability and the ICT Centre of Excellence Program.

Conceptually, the cost function is sufficiently simple that the non-linearity of the problem is entirely coded in the nature of the geometric constraint. Moreover, the geometric constraint is sufficiently symmetric that using local coordinates associated with the natural geometric structure will provide a good parameterisation of the problem in order to compute an iterative update step. A number of very efficient algorithms have been developed based on this intuition.

In this paper we consider optimization problems motivated by the same class of linear algebra problems discussed above, however, we develop by exploiting the extrinsic geometry of the constraint sets. That is, we wish to consider the embedding structure of the constraint set in some overarching space on which a cost function may be defined that specialises to the specific cost on the constraint set. In a sense we approach the problem from the perspective of classical constrained optimization, however, it is our goal to base the algorithms developed on the twin principals of symmetry of the constraint set and simplicity of the cost.

To this extent we propose to approximate the constraint set (in the case where it is a hypersurface in the embedding space) by a quadric, solve the optimization problem on the quadric, and project back onto the hypersurface in each iteration step. Clearly, this will only lead to a competitive algorithm where each of these steps is easily computable.

II. A SIMPLE EXAMPLE

A simple example that can be used to motivate the approach is optimization of the Rayleigh quotient on a sphere. Note that the sphere is a hypersurface and a quadric, i.e. it equals its quadric approximation.

Problem 1: Let $A = A^T \in \mathbb{R}^{n\times n}$ be a symmetric matrix. Define the Rayleigh quotient cost as

$$\phi(x) = x^T A x, \quad x \in \mathbb{R}^n. \tag{1}$$

Define the $n-1$ sphere to be

$$S^{n-1} = \{x \in \mathbb{R}^n \mid x^T x = 1\}. \tag{2}$$

The problem considered is to compute

$$x_* = \arg \max_{x \in S^{n-1}} \phi(x)$$

Problem 1 corresponds to computing the eigenvector corresponding to the maximal eigenvalue of A.

A. A Newton iteration for Problem 1 in intrinsic coordinates

It is well known that applying a Riemannian manifold version of the Newton iteration (i.e. a Newton iteration in Riemannian normal coordinates) to this problem yields a variant of the Rayleigh Quotient Iteration (RQI), see [10], [5], [1], [8].

B. An extrinsic geometric formulation of the problem

Consider now solving the problem

$$x_* = \arg\max_{x\in\mathbb{R}^n} \phi(x)$$

subject to the constraint

$$\psi(x) = x^T x - 1 = 0.$$

As pointed out before, the constraint set is a quadric, hence there is no need for an approximation step, and also not for a subsequent projection step in this case. We end up with a one step algorithm whose single step can be computed using Lagrange multiplier theory. The critical points of the Lagrange function $\mathcal{L}(x,\lambda) = \phi(x) + \lambda\psi(x)$ are given by

$$D\phi(x) = -\lambda D\psi(x)$$

for $\lambda \in \mathbb{R}$ and $x \in \mathbb{R}^n$ such that $\psi(x) = 0$.

That is one must solve

$$Ax = -\lambda x, \quad \text{subject to} \quad x^T x = 1.$$

Not surprisingly, the solution of this update step is equivalent to directly solving the eigenvalue problem. It is a nice indication that the method is theoretically well posed, however, there is obviously no computational advantage in this formulation.

C. A naive simplification of the extrinsic geometric formulation of the problem

To simplify the Lagrange function we consider replacing the cost ϕ by a linear approximation of the cost at the point x_0:

$$\phi_{x_0}(x) = x_0^T A x_0 + 2x_0^T A(x - x_0) = 2x_0^T A x - x_0^T A x_0.$$

In this case the critical points of the Lagrange function $\mathcal{L}(x,\lambda) = \phi_{x_0}(x) + \lambda\psi(x)$ are

$$2Ax_0 = -\lambda x, \quad \text{subject to} \quad x^T x = 1.$$

Note that this leads to the new estimate

$$x_+ = \frac{Ax_0}{\|Ax_0\|}.$$

The algorithms derived from this is the power method

$$x_{k+1} = \frac{Ax_k}{\|Ax_k\|}.$$

Clearly, this will be a successful algorithm for computing the maximal eigenvector, however, it is hardly a highly competitive algorithm.

D. A cost that respects the quotient structure of S^{n-1}

What happens if we consider the full Rayleigh quotient

$$\phi(x) = \frac{x^T A x}{x^T x}?$$

The Lagrange function is given by $\mathcal{L}(x,\lambda) = \phi(x) + \lambda\psi(x)$ as before.

The explicit constrained optimization problem with lagrange multipliers is equivalent to solving

$$2\left(A - \frac{x^T A x}{x^T x} I\right) x = -\lambda x, \quad \text{subject to} \quad x^T x = 1.$$

This is nothing particularly interesting – until we linearize it. We get

$$\phi_{x_0}(x) = 2\frac{x_0^T A x}{x_0^T x_0} - 2\frac{x_0^T A x_0}{x_0^T x_0}\cdot\frac{x_0^T x}{x_0^T x_0} + \frac{x_0^T A x_0}{x_0^T x_0}$$

and hence the critical points of the Lagrange function become

$$2\left(A - \frac{x_0^T A x_0}{x_0^T x_0} I\right) x_0 = -\lambda x, \quad \text{subject to} \quad x^T x = 1,$$

for any x_0 that satisfies $x_0^T x_0 = 1$. Obviously, the resulting update rule leads directly to the RQI!

E. Another practical algorithm for the extrinsic approach

Consider again the case where the cost is

$$\phi(x) = x^T A x$$

and the constraint is

$$\psi(x) = x^T x - 1 = 0.$$

The Lagrange function is

$$\mathcal{L}(x,\lambda) = \phi(x) + \lambda\psi(x).$$

The Differential and the Hessian of $\mathcal{L}$ (i.e. the bordered Differential and Hessian of ϕ under the constrained $\psi = 0$) are given by

$$D\mathcal{L} = \begin{pmatrix} 2(Ax + \lambda x) \\ x^T x - 1 \end{pmatrix} \text{ and } H\mathcal{L} = \begin{pmatrix} A + \lambda I & x \\ x^T & 0 \end{pmatrix}.$$

The Newton update for $\mathcal{L}$ on $\mathbb{R}^{n+1}$ is given by

$$x_{k+1} = x_k - \Pi_x H\mathcal{L}(x_k,\lambda_k)^{-1} D\mathcal{L}(x_k,\lambda_k)$$

and

$$\lambda_{k+1} = \lambda_k - \Pi_\lambda H\mathcal{L}(x_k,\lambda_k)^{-1} D\mathcal{L}(x_k,\lambda_k),$$

where Π_x and Π_λ denote the appropriate projections. This iteration should converge to a critical point of $\mathcal{L}$. Again, such points are characterised by

$$x^T x = 1$$

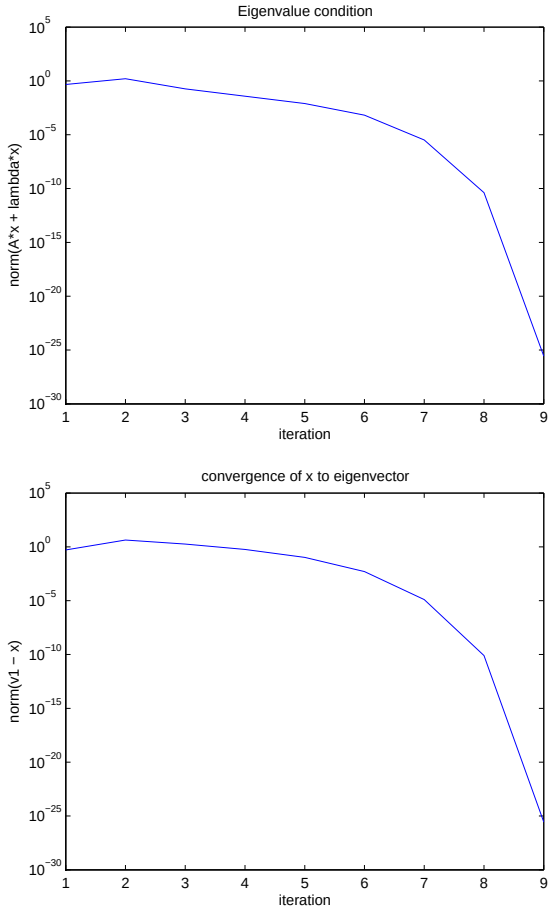
and

$$Ax = -\lambda x,$$

that is x is an eigenvector and

$$\lambda = -\frac{x^T A x}{x^T x}.$$

Here are some simulation results using MATLAB.



Evolution for the Newton algorithm choosing a random initial condition $x_0 \in S^{n-1}$, $\lambda_0 = -\frac{x_0^T A x_0}{x_0^T x_0}$ and $A = \mathrm{diag}(1,2,3)$. The first plot shows the convergence of the eigenvalue condition $Ax = -\lambda x$ and the second shows direct convergence of $x_k \to v_1$.

Some comments are in order:

- As the algorithm converges it is clear that λ converges to the Rayleigh quotient at least quadratically. The upper left part of the Newton algorithm starts to look something like the RQI.
- The convergence appears to be quadratic. The pure RQI is cubic.
- However, it is clear that this algorithm is not the RQI.

III. A GENERAL ALGORITHM

The above approach is a logical solution of the problem, however, in a more general case the curvature will change at every point in the manifold. Thus we need to have a methodology that re-projects to the manifold after each iteration.

Given a manifold $M = \{x \in \mathbb{R}^n \mid \psi(x) = 0\}$ and a cost $\phi : \mathbb{R}^n \to \mathbb{R}$. Assume that there is a computationally efficient projection operation

$$\Pi : \mathbb{R}^n \to M.$$

Define the Lagrange function by

$$\mathcal{L} = \phi(x) + \lambda\psi(x).$$

Then the basic algorithm is
Algorithm 1:

1) Given initial conditions $x_0 \in M$.
2) Compute the best estimate of λ based on the requirement that the normal component of the Differential of the Lagrange function is zero, i.e. solve for λ_0

$$\langle D\psi(x_0), \Pi_x D\mathcal{L}(x_0, \lambda_0)\rangle = 0. \tag{3}$$

3) Initialise $x_k = x_0$ and $\lambda_k = \lambda_0$ and iterate the following routine to convergence:
   a) Compute the quadratic approximation of the cost and the manifold at x_k,

$$\phi(x) = m^{\phi}_{x_k}(x) + O(|x - x_k|^3)$$

   and

$$\psi(x) = m^{\psi}_{x_k}(x) + O(|x - x_k|^3).$$

   b) Compute an approximate solution x_{k+1} to the optimization problem associated with minimizing $m^{\phi}_{x_0}$ subject to $m^{\psi}_{x_0} = 0$.
   The present proposal is to use a Newton step of the Lagrange multiplier update in (x_k, λ_k)

$$\begin{pmatrix} x_{k+1} \\ \lambda_k \end{pmatrix} = \begin{pmatrix} x_k \\ \lambda_k \end{pmatrix} - H\mathcal{L}(x_k, \lambda_k)^{-1} D\mathcal{L}(x_k, \lambda_k).$$

   It is entirely possible that a different approach could be used effectively here. Indeed, it would be equally possible to use several iterations of the Newton method at this point to improve the x_{k+1} estimate if desired.
   c) Re-project x_{k+1} onto M

$$\bar{x}_{k+1} = \Pi(x_{k+1})$$

   and discard λ_{k+1}.
   d) Compute a new $\bar{\lambda}_{k+1}$ based on the requirement that the normal component of the Differential of the Lagrange function is zero, i.e. solve for $\bar{\lambda}_{k+1}$

$$\langle D\psi(\bar{x}_{k+1}), \Pi_x D\mathcal{L}(\bar{x}_{k+1}, \bar{\lambda}_{k+1})\rangle = 0.$$

   e) Check for numerical convergence. If not set

$$x_k = \bar{x}_{k+1}, \quad \lambda_k = \bar{\lambda}_{k+1}$$

   and continue the iteration.

For the eigenvalue computation we have the Hessian and Differential given as computed above. Note that the condition

$$\langle D\psi(\bar{x}_{k+1}), \Pi_x D\mathcal{L}(\bar{x}_{k+1}, \bar{\lambda}_{k+1})\rangle = 0$$

for the eigenvalue computation leads to

$$\bar{x}_{k+1}^T \left(A\bar{x}_{k+1} + \bar{\lambda}_{k+1}\bar{x}_{k+1}\right) = 0$$

or

$$\bar{\lambda}_{k+1} = -\frac{\bar{x}_{k+1}^T A \bar{x}_{k+1}}{\bar{x}_{k+1}^T \bar{x}_{k+1}},$$

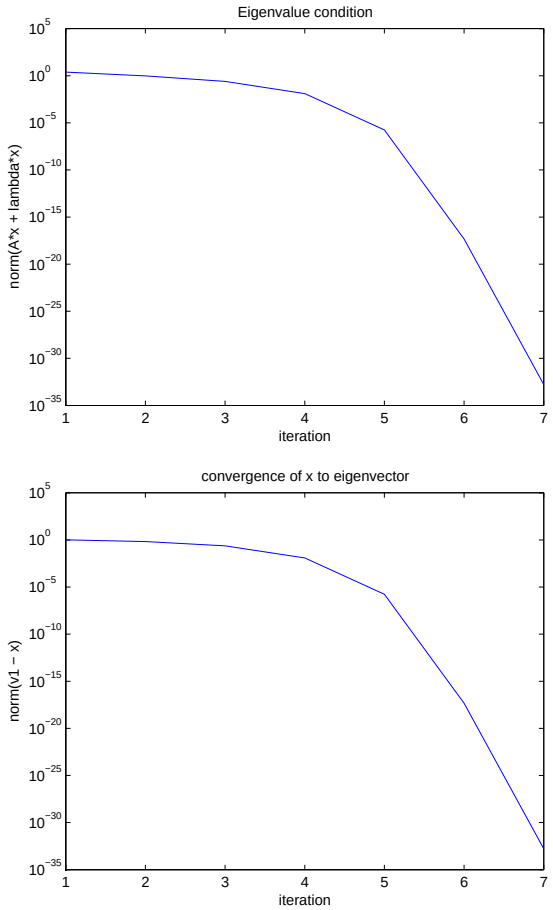
where $\bar{x}_{k+1}^T \bar{x}_{k+1} = 1$ due to the projection, but the form of the equation is useful to see in general.

The Newton update for the re-projected scheme can be written as the solution of the iteration

$$(A + \lambda_k I + x_k x_k^T A) x_{k+1} = A x_k + \lambda_k x_k,$$
$$x_k^T x_{k+1} = 0.$$

It is clear that this is not quite the same as the RQI, although it appears that it shares some of the same major characteristics.

A simulation shows that this algorithm displays cubic convergence:



Evolution for the projected external geometry algorithm using a single Newton update. A random initial condition was chosen $x_0 \in S^{n-1}$, $\lambda_0 = -\frac{x_0^T A x_0}{x_0^T x_0}$ and $A = \mathrm{diag}(1, \ldots, N)$ for $N = 10$. The first plot shows the convergence of the eigenvalue condition $Ax = -\lambda x$ and the second shows direct convergence of $x_k \to v_1$.

IV. CONCLUSIONS

We have introduced a new class of algorithms for the optimization of a smooth cost function defined on a hypersurface. For the Rayleigh quotient on the sphere the resulting algorithm displays cubic convergence and shares properties with the RQI. We have seen that the power method results from applying our scheme to a linearization of the extrinsic cost function, whereas the RQI results from the linearization of a geometrically more meaningful cost function. In both cases the optimization problem on the quadric approximation of the constraint set could be solved explicitely. Doing just one Newton step in the bordered space instead yields our new algorithm. As an interesting side note, we see that optimizing the quadratic cost over a linearization of the constrained set (this is effectively what the intrinsic Newton method does) gives the same result – the RQI – as optimizing a proper linearization of the cost over a quadric approximation of the constraint set.

REFERENCES

[1] P.-A. Absil, R. Mahony, R. Sepulchre, and P. Van Dooren. A Grassmann-Rayleigh quotient iteration for computing invariant subspaces. *SIAM Review*, 44(1):57-73, 2002.

[2] A. M. Bloch. Steepest descent, linear programming and Hamiltonian flows. *Contemporary Math.*, 114:77-88, 1990.

[3] A. M. Bloch, H. Flaschka, and T. Ratui. A convexity theorem for isospectral sets of Jacobi matrices in a compact Lie algebra. *Duke Math. J.*, 61:41-65, 1990.

[4] R. W. Brockett. Smooth dynamical systems which realize arithmetical and logical operations. In *Three Decades of Mathematical Systems Theory*, number 135 in Lecture Notes in Control and Information Sciences, pages 19-30. Springer-Verlag, 1989.

[5] A. Edelman, T. A. Arias, and S. T. Smith. The geometry of algorithms with orthogonality constraints. *SIAM Journal on Matrix Analysis and Applications*, 20(2):303-353, 1998.

[6] H. Flashka. The Toda lattice, II. Existence of integrals. *Physical Review B*, 9(4):1924-1925, 1974.

[7] U. Helmke and J. B. Moore. Singular value decomposition via gradient flows. *Systems and Control Letters*, 14:369-377, 1990.

[8] K. Hüper and J. Trumpf. Newton-like methods for numerical optimization on manifolds. *Proceedings of the Asilomar Conference on Signals, Systems, and Computers*, pages 136–139, Pacific Grove, CA, USA, 2004.

[9] J. Moser. Finitely many mass points on the line under the influence of an exponential potential - an integrable system. In J. Moser, editor, *Dynamic Systems Theory and Applications*, pages 467-497, New York, 1975. Springer-Verlag.

[10] M. Shub. Some remarks on dynamical systems and numerical analysis. In *Dynamical systems and partial differential equations (Caracas, 1984)*, pages 69–91, Caracas: Univ. Simon Bolivar, 1986.

[11] W. W. Symes. The QR algorithm and scattering for the finite nonperiodic Toda lattice. *Physica 4D*, pages 275-280, 1982.